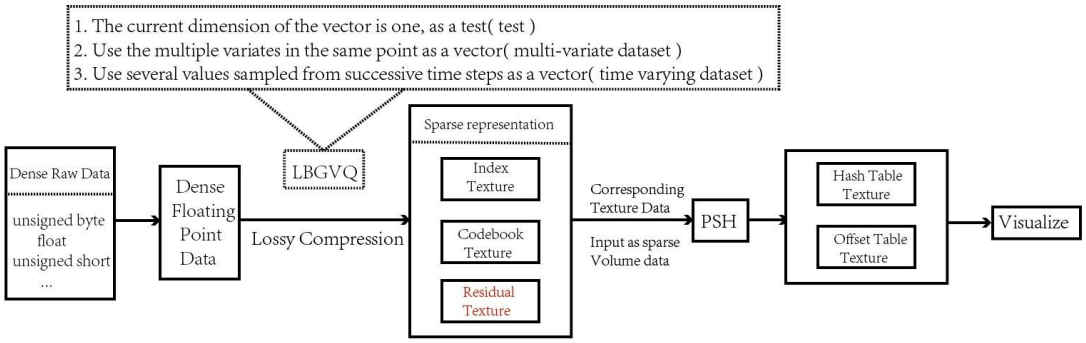
Weekly report

I constructed a pipeline for compression of volume data, including both the type of unsigned byte and float as well as unsigned short.

It is currently a prototype without refinement and verifying. The basic idea and pipeline is showed as below.



First, all these kinds of volume data are taken as input and followed by a floated process. The produced dense floating-point volume data will be handled using LBGVQ, a lossy compression strategy. In the current time, the vector dimension is set to be one as a test. In the future, the multiple variates in the same point and values sampled from several successive time steps will be considered as a vector in the case of multi-variate data set and time varying dataset respectively. After the compression, we will get a sparse representation of the original input, namely, the index texture, codebook texture and maybe a residual texture which is not sure to exist or not.

In the PSH procedure, the input sparse volume data should be the corresponding texture data generated in the previous stage. Finally, a hash table and an offset table will be the expression of the original data used as input of the visualize stage.

Now, all the stages in the pipeline are implemented and tested separately. In the test case, the Bucky volume data(32, 32, 32) is taken as the input of the PSH and visualize stage and the result is correct as the previous weekly report described. As for the first stage(LBGVQ), I use the climate data climateTestTMP360x181x41(floating-point volume data), which is original from t639_grib1. It is extracted by using wgrid.exe first, followed by a convert function developed by me. The result of VQ is depicted as following.

```
Input data file           : climate360_181_41.raw
Number of training vectors : 2671560
Dimension                 : 1
Rate of VQ (bit/sample)   : 8.000000
Filename of VQ codebook   : codebook.dat
Filename of results       : results.dat


cb_size=   2  Ave. Dist = 5985.245605 SNR = -37.770820 drel = 1670775308288.000000
cb_size=   2  Ave. Dist = 3151.370117 SNR = -34.984994 drel = 0.899252
cb_size=   2  Ave. Dist = 963.995850 SNR = -29.840752 drel = 0.000000
cb_size=   4  Ave. Dist = 950.705505 SNR = -29.780460 drel = 10518504931328.000000
cb_size=   4  Ave. Dist = 212.527817 SNR = -23.274158 drel = 0.000573
cb_size=   8  Ave. Dist = 206.947968 SNR = -23.158612 drel = 48321326481408.000000
cb_size=   8  Ave. Dist = 66.224243 SNR = -18.210170 drel = 0.000858
cb_size=  16  Ave. Dist = 62.940651 SNR = -17.989312 drel = 158879832866816.000000
cb_size=  16  Ave. Dist = 18.241329 SNR = -12.610565 drel = 0.000597
cb_size=  32  Ave. Dist = 16.594950 SNR = -12.199759 drel = 602592971325440.000000
cb_size=  32  Ave. Dist = 5.107237 SNR = -7.081860 drel = 0.007758
...

cb_size=  32  Ave. Dist = 4.789858 SNR = -6.803226 drel = 0.000337
cb_size=  64  Ave. Dist = 3.950944 SNR = -5.967008 drel = 2531040870531072.000000
cb_size=  64  Ave. Dist = 1.263549 SNR = -1.015921 drel = 2.126862
cb_size= 128  Ave. Dist = 0.852991 SNR = 0.690554 drel = 11723448241881088.000000
cb_size= 128  Ave. Dist = 0.316969 SNR = 4.989828 drel = 1.691085
cb_size= 256  Ave. Dist = 0.144662 SNR = 8.396441 drel = 69126433477230592.000000
cb_size= 256  Ave. Dist = 0.079728 SNR = 10.983914 drel = 0.000402

Ave. Distortion of VQ = 0.079728
     SNR        of VQ = 10.983914
```

Next week, I will try to merge all these independent programs together and verify the result based on some different volume data to check both the effectiveness and efficiency. If everything goes well, I think the basic work is done and this is also the first goal of the current project.

Reference.

1. Compression Domain Volume Rendering. IEEE VIS 2003
2. High Performance Visualization of Time Varying Volume Data over a Wide Area Network.
3. Lossless Compression of Volume Data.
4. S3Dc: A 3Dc-based Volume Compression Algorithm. CEIG 08.
5. A Decompression Pipeline for Accelerating Out-of-Core Volume Rendering of Time-Varying Data. Computers & Graphics 2008.